

Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction

Hermann G. Matthies, Jan Steindorf
Institute of Scientific Computing
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-04

July 2002



Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction

Hermann G. Matthies, Jan Steindorf
Department of Computer Science
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-04
July 2002

Contributed to a special Issue of *Computers and Structures*.

Location

Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Strasse 65
D-38106 Braunschweig

Postal Address

Institut für Wissenschaftliches Rechnen
Technische Universität Braunschweig
D-38092 Braunschweig
Germany

Contact

Phone: +49-(0)531-391-3000
Fax: +49-(0)531-391-3003
E-Mail: wire@tu-bs.de

Copyright

2002 ©Institut für Wissenschaftliches Rechnen
Technische Universität Braunschweig

Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction

Hermann G. Matthies Jan Steindorf
Institute of Scientific Computing
Technische Universität Braunschweig
Brunswick, Germany July 2002

Abstract

Numerical simulation of fluid-structure interaction is often attempted in the context of partitioned methods, where already existing solvers for fluid or structure alone are used jointly. Mostly this is done by exchanging information from time step to time step in an alternating fashion. These weak coupling methods are explicit and hence suffer from possible instabilities. Therefore often strong coupling — where equilibrium is satisfied jointly between fluid and structure in each time step — is desired; the simplest computational procedure is similar to the time stepping an alternating iteration. We show why also this approach may experience difficulties, and how they may be circumvented with block-Newton methods, still in the partitioned framework, by only using the solvers of the subproblems fluid and structure.

Keywords: fluid-structure-interaction, partitioned methods, strong coupling, block-Newton methods

1 Introduction

Fluid-Structure Interaction problems often show strong interplay between the fluid and the structure [1, 12], be it in the design of aircraft [2, 3], helicopters [4, 5], in the consideration of sloshing in tanks [6], or be it in biomechanical applications [7] — see also the examples in [14].

One possibility is to develop new software and solution methods for each of these coupled applications, as undoubtably will happen in some areas. This is referred to as a *monolithic* approach [8], or sometimes as the *direct* method [16, 15, 17]. On the other hand, we shall assume here that the methods and software systems which have been developed for either fluid or structural applications will continue to be used. Therefore we consider *partitioned* methods [9, 10, 11] — also known as *iterative* methods [16, 15, 17] for fluid-structure interaction, i.e. separate solvers are used for the fluid and the structure [18, 24, 25].

For stability reasons, often a fully implicit formulation has to be used [15, 17, 6]. In this approach, we have to solve a large system of nonlinear equations with

the use of the (iterative) solvers for the subsystems. Commonly this is performed with block-Jacobi, block-Gauss-Seidel or related relaxation methods [19]. These simple methods do not always converge [20, 21, 22, 23, 24, 25]. We will introduce here a superior approximative block-Newton method [26, 22, 23, 24, 25].

2 Partitioned Formulation

We assume the fluid to be modeled adequately as an incompressible Newtonian fluid, satisfying the appropriate Navier-Stokes equation. As we want to couple this with a moving structure, we have to allow for a moving boundary. We take account of this by formulating the Navier-Stokes equation in an Arbitrary Lagrangean-Eulerian (ALE) [27] framework in the moving fluid domain Ω_f :

$$\begin{aligned} (1) \quad & \varrho_f (\dot{v} + ((v - \dot{x}) \cdot \nabla)v) - \operatorname{div} \sigma + \nabla p = f, \\ (2) \quad & 2\sigma = \nu(\nabla v + (\nabla v)^T), \quad \operatorname{div} v = 0. \end{aligned}$$

The boundary $\partial\Omega_f$ we assume to be divided into three disjoint parts $\partial\Omega_f = \Gamma_v \cup \Gamma_q \cup \Gamma_c$ where on Γ_v the velocity is prescribed, on Γ_q the traction is given, and Γ_c is the coupling boundary, where the coupling conditions will be specified below. Here ϱ_f is the fluid density, v the velocity, σ the viscous stress, the superimposed dot signifies the partial derivative w.r.t. time t , x is the position of the reference ALE-coordinate system and \dot{x} its velocity. The fluid shear viscosity is denoted by ν , p is the pressure, g the body force in the fluid, and the differential operators are in the spatial frame.

The structure we assume to be modeled by a neo-Hookean material, hence the equilibrium equation takes the following form in a Lagrangean framework in a fixed domain Ω_s :

$$\begin{aligned} (3) \quad & \varrho_s \ddot{u} - \operatorname{DIV}(FS) = f, \quad F = \operatorname{GRAD} u \\ (4) \quad & S = \lambda(\operatorname{tr} E)I + 2\mu E, \quad 2E = (C - I), \quad C = F^T F \end{aligned}$$

The boundary $\partial\Omega_s$ we again assume divided into three disjoint parts $\partial\Omega_s = \Gamma_u \cup \Gamma_t \cup \Gamma_c$, where on Γ_u the displacements u are prescribed, on Γ_t the tractions, and Γ_c is the coupling boundary with the fluid. The other quantities are the structure density ϱ_s , the displacement gradient F , the second Piola-Kirchhoff stress S , and the body load f . The Lamé moduli are denoted by λ and μ , and E is the Lagrange-Green strain, derived from the Cauchy-Green tensor C , and the differential operators are in the material frame.

On the coupling boundary Γ_c let a unique normal n be defined in the spatial frame. Then the coupling conditions may be expressed as requiring the velocities

to coincide at the location $x(t) = x_0 + u(x_0, t)$

$$(5) \quad v(x(t), t) = \dot{u}(x_0, t),$$

and the tractions from fluid and structure to balance each other:

$$(6) \quad (\sigma - pI) \cdot n = -\frac{1}{J} F S F^T \cdot n.$$

3 Discrete Formulation

We shall further assume that the fluid — described by Eq. (1) — and the structure — described by Eq. (3) — have been discretised; each one may use her favourite discretisation, e.g. finite elements, finite volumes etc., e.g. [13].

Up to now we have omitted to say how to move the reference coordinate system in the fluid domain. Several possibilities exist [28, 7, 29, 6], here we have followed [30] and modeled the connections between the nodes in the fluid domain as elastic springs. This fictitious inertia-free elastic body has displacement loading from the moving structure, on the outer boundaries it is fixed. The traction balance Eq. (6) introduces additional forces on the fluid, and the reaction force on the solid. Hence for the fluid the complete discrete equations of motion are [22]:

$$(7) \quad \mathbf{M}_f \dot{\mathbf{v}} + \mathbf{N}(\mathbf{v} - \dot{\mathbf{x}}) \mathbf{v} + \mathbf{K}_f \mathbf{v} + \mathbf{B}_f \mathbf{p} = \mathbf{g} + \mathbf{T}_f^T \boldsymbol{\tau}$$

$$(8) \quad \mathbf{B}_f^T \mathbf{v} = \mathbf{0}$$

$$(9) \quad \mathbf{K}_g \mathbf{x} = \mathbf{A} \mathbf{u}.$$

The terms in Eq. (7) are the discrete analogues of those in Eq. (1), the term \mathbf{g} includes the prescribed boundary stresses, and the additional term $\mathbf{T}_f^T \boldsymbol{\tau}$ comes from the interaction with the structure, where $\boldsymbol{\tau}$ is the stress on the fluid-structure-interface. In Eq. (8) we see the discrete form of the incompressibility condition, and Eq. (9) describes the movement \mathbf{x} of the fluid domain, driven by the structure displacements \mathbf{u} .

Similarly for the structure we obtain

$$(10) \quad \mathbf{M}_s \ddot{\mathbf{u}} + \mathbf{K}_s(\mathbf{u}) \mathbf{u} = \mathbf{f} - \mathbf{T}_s^T \boldsymbol{\tau}.$$

To this we have to add the discrete coupling condition for the velocities,

$$(11) \quad \mathbf{T}_f \mathbf{v} = \mathbf{T}_s \dot{\mathbf{u}}.$$

This combined set is a system of differential algebraic equations (DAEs) in the time variable t of index 2, as can be easily verified. In order to allow an easier numerical treatment and to make them fit the more general formulation in section 4, they will be converted to index 1 by differentiation [22].

Differentiating Eq. (8) once gives $\mathbf{B}_f^T \dot{\mathbf{v}} = \mathbf{0}$. As \mathbf{M}_f is non-singular, we may solve for $\dot{\mathbf{v}}$ from Eq. (7) and insert this into the above relation, giving

$$(12) \quad \mathbf{B}_f^T \mathbf{M}_f^{-1} (\mathbf{g} + \mathbf{T}_f^T \boldsymbol{\tau} - \mathbf{N}_f(\mathbf{v} - \dot{\mathbf{x}}) \mathbf{v} - \mathbf{K}_f \mathbf{v} - \mathbf{B}_f \mathbf{p}) = \mathbf{0}.$$

Similarly, Eq. (9) for the grid movement has to be differentiated

$$(13) \quad \mathbf{K}_g \dot{\mathbf{x}} - \mathbf{A} \mathbf{w} = \mathbf{0},$$

where we have introduced the new variable $\mathbf{w} = \dot{\mathbf{u}}$. With this definition the structural Eq. (10) can be reduced to first order:

$$(14) \quad \mathbf{M}_s \dot{\mathbf{w}} + \mathbf{K}_s(\mathbf{u}) \mathbf{u} = \mathbf{f} - \mathbf{T}_s^T \boldsymbol{\tau}.$$

4 Abstract Coupling Algorithms

The development in the preceeding section 3 enables us to write the coupled system in the form [21]:

$$(15) \quad \dot{\mathbf{y}}_1 = \mathbf{f}_1(\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1)$$

$$(16) \quad \mathbf{0} = \mathbf{g}_1(\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{v}),$$

and

$$(17) \quad \dot{\mathbf{y}}_2 = \mathbf{f}_2(\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_2)$$

$$(18) \quad \mathbf{0} = \mathbf{g}_2(\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_2, \mathbf{v}),$$

and the global coupling condition

$$(19) \quad \mathbf{0} = \mathbf{h}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{z}_1, \mathbf{z}_2).$$

The global DAE, as well as each local DAE, whether by itself or combined with the global coupling condition has to be an index 1 system. This means that the matrices

$$(20) \quad D_{z_i} \mathbf{g}_i, \quad \begin{bmatrix} D_{z_i} \mathbf{g}_i & D_v \mathbf{g}_i \\ D_{z_i} \mathbf{h} & \mathbf{0} \end{bmatrix}, \quad i = 1, 2, \quad \begin{bmatrix} D_z \mathbf{g} & D_v \mathbf{g} \\ D_z \mathbf{h} & \mathbf{0} \end{bmatrix}$$

have to be regular, where D_q is the partial derivative w.r.t. q .

For our concrete application, the abstract Eqs.(15–19) may be identified with the following terms [22]:

$$(21) \quad \mathbf{y}_1 = \begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix}, \quad \mathbf{z}_1 = \begin{bmatrix} \mathbf{b} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{f}_1 = \begin{bmatrix} \mathbf{b} \\ \boldsymbol{\xi} \end{bmatrix}, \quad \mathbf{v} = \boldsymbol{\tau}$$

$$(22) \quad \mathbf{g}_1 = \begin{bmatrix} \mathbf{M}_f \mathbf{b} + \mathbf{N}_f(\mathbf{v} - \boldsymbol{\xi}) \mathbf{v} + \mathbf{K}_f \mathbf{v} + \mathbf{B}_f \mathbf{p} - \mathbf{g} - \mathbf{T}_f^T \boldsymbol{\tau} \\ -\mathbf{B}_f^T \mathbf{M}_f^{-1} (-\mathbf{N}_f(\mathbf{v} - \boldsymbol{\xi}) \mathbf{v} - \mathbf{B}_f \mathbf{p} - \mathbf{K}_f \mathbf{v} + \mathbf{g} + \mathbf{T}_f^T \boldsymbol{\tau}) \\ \mathbf{K}_g \boldsymbol{\xi} - \mathbf{A} \mathbf{w} \end{bmatrix}$$

for the fluid,

$$(23) \quad \mathbf{y}_2 = \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix}, \quad \mathbf{z}_2 = \mathbf{a}, \quad \mathbf{f}_2 = \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix}, \quad \mathbf{v} = \boldsymbol{\tau}$$

$$(24) \quad \mathbf{g}_2 = \mathbf{M}_s \mathbf{a} + \mathbf{K}_s(\mathbf{u})\mathbf{u} - \mathbf{f} + \mathbf{T}_s^T \boldsymbol{\tau}$$

for the structure, and

$$(25) \quad \mathbf{h} = \mathbf{T}_f \mathbf{b} - \mathbf{T}_s \mathbf{a}$$

for the coupling condition.

This abstract setting allows us a simple formulation of different coupling algorithms. Let us remark that in the chosen formulation the differential equation only gives the definition of the various variables, whereas the equilibrium equations have gone into the algebraic conditions. What is still left is to verify the index 1 condition Eq. (20). This is a lengthy calculation [22], but the index 1 conditions are indeed satisfied as long as \mathbf{K}_g , \mathbf{M}_f , and \mathbf{M}_s are regular — in our case they even can be assumed to be symmetric positive definite.

Let us assume that we have integrators for each system separately. We express this by saying that given the state at time t_n , we have for each system separately a function Φ_j , $j = 1, 2$ which will advance the state — or an approximation thereof — to the end of the time increment at time t_{n+1} . For the sake of simplicity we shall confine ourselves to the case where the time step in each subsystem is the same. Hence, for subsystem 1, we may say that given the values $(\mathbf{y}_1^{(n)}, \mathbf{z}_1^{(n)})$ and the functions $(\mathbf{y}_2, \mathbf{z}_2, \mathbf{v})$ over the interval $[t_n, t_{n+1}]$, we obtain

$$(26) \quad (\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}) = \Phi_1(\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{y}_1^{(n)}, \mathbf{z}_1^{(n)}, \mathbf{y}_2, \mathbf{z}_2, \mathbf{v}).$$

We have written this equation in fixed point form for $(\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)})$, assuming that this is also the iterative method to solve for the new values in case the single system method is implicit.

Similarly, for subsystem 2 we have an analogous function, we only have to exchange indices in Eq. (26):

$$(27) \quad (\mathbf{y}_2^{(n+1)}, \mathbf{z}_2^{(n+1)}) = \Phi_2(\mathbf{y}_2^{(n+1)}, \mathbf{z}_2^{(n+1)}, \mathbf{y}_2^{(n)}, \mathbf{z}_2^{(n)}, \mathbf{y}_1, \mathbf{z}_1, \mathbf{v}).$$

From these single system integrators, different combinations may be built in order to integrate the global system.

5 Partitioned Solution Strategies

Probably the simplest and the best known are the staggered schemes, also referred to as weak or loose coupling. We first have to decide where we want to put the

global constraint Eq. (25). Usually this is put onto the “fluid-side”, i.e. we transfer the structural displacements, velocities, and accelerations onto the fluid domain, and not vice versa. On the other hand we then apply the fluid stress to the structure. We shall retain the denotation Φ_1 also for this extended solution function, which now solves

$$(28) \quad (\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{v}^{(n+1)}) = \Phi_1(\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{v}^{(n+1)}, \mathbf{y}_1^{(n)}, \mathbf{z}_1^{(n)}, \mathbf{v}^{(n)}, \mathbf{y}_2, \mathbf{z}_2).$$

The simplest staggered scheme then is for subsystem 1 to assume the time evolution of the state variables $(\mathbf{y}_2, \mathbf{z}_2)$ to be constant over the interval $[t_n, t_{n+1}]$, i.e. identically equal to $(\mathbf{y}_2^{(n)}, \mathbf{z}_2^{(n)})$, and then solve Eq. (28). Similarly for subsystem 2 we assume $(\mathbf{y}_1, \mathbf{z}_1, \mathbf{v})$ over the time increment to be constant and identically equal to the initial values $(\mathbf{y}_1^{(n)}, \mathbf{z}_1^{(n)}, \mathbf{v}^{(n)})$, and with this we can solve Eq. (27). To sum up, information is only exchanged between the subsystems after the increment has been performed; otherwise inside the increment, each subsystem is solved in parallel independently of the other. It is quite easy to see that this scheme will only give first order accuracy in time, no matter how accurate the single system integrators are. This could be improved by a higher order extrapolation, but this will have a negative impact on the stability characteristics.

There are many variations on this staggering theme [20, 31], but they all share the same characteristic in that they are explicit when looked at as time-integrators although the integrators for the individual systems may be implicit. They also suffer from the same limitations as other explicit schemes in that they are not unconditionally stable, although some remedies exist for the linear stability. Often the time step limitations imposed by this explicit part are so severe that one wants to use an implicit method on the global level. Additionally, if global constraints have to be satisfied exactly, explicit methods are of no great use; global invariants such as energy conservation will almost certainly be violated by explicit schemes. All this makes the use of implicit methods desirable [20, 23, 24]. They offer the possibility to achieve the same results as monolithic methods [8], but with partitioned software. Satisfying an implicit condition globally is often referred to as strong or tight coupling. Additionally, this strong coupling allows better stability through energy conservation [10, 6, 32]. Another motivation for the use of implicit methods on the global level is that we are solving a DAE, and if we want to ensure that the constraints are satisfied, then it is advisable to use implicit methods.

As implicit methods will usually be iterative, we first need a starting value for the iteration through a process of extrapolating the solution into the new time step. Assuming that we know the values $(\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{y}_2^{(n+1)}, \mathbf{z}_2^{(n+1)})$ and all the proceeding ones, we may define some kind of interpolation for the time interval $[t_n, t_{(n+1)}]$. The simplest possibilities are the constant values

$(\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{y}_2^{(n+1)}, \mathbf{z}_2^{(n+1)})$, or the linear interpolation between these and the values at t_n . With this kind of extrapolation, we may again enter a process similar to Eqs.(27, 28). But as the values at t_{n+1} were merely assumed and not known, this must now be regarded as an iterative process; the simplest again being certainly that one where the simple staggering idea is now performed iteratively inside one time increment.

That means that given some — assumed and approximate — values at time t_{n+1} , we perform one of the afore mentioned interpolation procedures, and then the process on Eqs.(27, 28) is carried out, i.e. each subsystem is solved independently, in parallel — of course, carrying this out in parallel is merely an option and not a requirement. This gives new, approximate values for the variables at t_{n+1} , and the whole process may now be performed in an iterative fashion until convergence.

This simple way of solving the global coupled system is easily recognised as some kind of nonlinear block-Jacobi method. We remember that there is no guarantee that this method will converge, even for starting values arbitrary close to the solution. Some kind of damping or under-relaxation or some other more sophisticated forms of pre-conditioning may be introduced in order to achieve convergence. This immediately brings to mind that the corresponding Gauss-Seidel process often converges faster [19]. It is easily constructed with the same ingredients. The only difference is that now we pick one subsystem — say subsystem 1 — to solve for first, then with these newly computed values we solve for subsystem 2, and this is repeated iteratively until convergence. As with the block-Jacobi process, we know that this process may also not converge, even for starting values arbitrarily close to the exact solution, and that the convergence behaviour depends on the order in which the subsystems are solved for. With only two coupled systems the ordering seems not so important and its influence could easily be tested, but for more than two coupled systems this is not feasible. It has been shown that it is possible to under-relax or dampen this process such that it is convergent for simple model problems [6, 32], and in [6, 11] it is discussed how to choose the relaxation parameter.

In this situation it is useful to note the following result [21, 33]:

Let

$$\alpha = \max_{t \in [0, T]} \| (D_{z_2} \mathbf{g}_2)^{-1} D_v \mathbf{g}_2 (D_{z_1} \mathbf{h} (D_{z_1} \mathbf{g}_1)^{-1} D_v \mathbf{g}_1)^{-1} D_{z_2} \mathbf{h} \|,$$

and let L be the Lipschitz-constant of the extrapolation. Assume that $\alpha < 1$, and that at least k iterations of the block-Gauss-Seidel scheme are performed, such that $L\alpha^k < 1$, and that $\Delta t = t_{n+1} - t_n$ is small enough. Then the global block-Gauss-Seidel method converges, and the error

$$\delta^{(n+1)} = \|\mathbf{y}^{(n+1)} - \mathbf{y}(t_{n+1})\| + \|\mathbf{z}^{(n+1)} - \mathbf{z}(t_{n+1})\| + \|\mathbf{v}^{(n+1)} - \mathbf{v}(t_{n+1})\|$$

is bounded by

$$(29) \quad \delta^{(n+1)} \leq C(\mu^{\max(0, k-2)} \delta^{(n)} \mathbf{y} + \mu^{k-1} \delta^{(n)} \mathbf{z}) + \varepsilon_1^{(n+1)} + \varepsilon_2^{(n+1)}.$$

Here $\varepsilon_1^{(n+1)}$ and $\varepsilon_2^{(n+1)}$ are the errors incurred by the single system integrators Eqs. (27,28) and $\delta^{(n)} \mathbf{y}$ and $\delta^{(n)} \mathbf{z}$ are the extrapolation errors, and $\mu = \alpha + O(\Delta t)$.

As long as Δt is small enough such that $\mu < 1$, the iteration will converge. We see that the contractivity constant α is crucial, and without $\alpha < 1$ the block-Gauss-Seidel method will not converge, no matter what the time step is; and the convergence depends strongly on the ordering of the subsystems in the block-Gauss-Seidel solution strategy.

Observe that if we perform enough iterations, essentially only the error components from the single system integrators remain. If we have $\varepsilon_1^{(n+1)} = O((\Delta t)^p)$ and $\varepsilon_2^{(n+1)} = O((\Delta t)^q)$ as convergence orders for the single system integrators, we obtain $\delta^{(n+1)} = O((\Delta t)^{\min(p,q)})$, in contrast to the staggering scheme where we only have $O(\Delta t)$.

In our view all this calls for strong or tight coupling methods which will converge unconditionally provided the time step is small enough; this then would roughly be a situation similar for the single system implicit integrator. With block-Gauss-Seidel methods there is no easy way of achieving this, although there are possibilities of preconditioning [21].

6 Block-Newton Methods

To have a more compact notation, we shall only consider one time step and introduce the fluid variables $\xi = (\mathbf{y}_1^{(n+1)}, \mathbf{z}_1^{(n+1)}, \mathbf{v}^{(n+1)})$ and the structure variables $\zeta = (\mathbf{y}_2^{(n+1)}, \mathbf{z}_2^{(n+1)})$. Then the above Eqs.(27,28) may be concisely written as

$$(30) \quad \xi = \Phi_1(\xi, \zeta)$$

$$(31) \quad \zeta = \Phi_2(\xi, \zeta).$$

It was observed in [26] that it is numerically advantageous to consider the system Eqs.(30,31), i.e. the iteration equations, rather than the equilibrium conditions; and to consider the use of Newton's method for the solution [26, 34, 35, 24, 25].

One step of Newton's method for this combined system Eqs.(30, 31) entails the solution of the following linear system at each iteration, with $\Delta \xi_k := \xi_{k+1} - \xi_k$ and $\Delta \zeta_k := \zeta_{k+1} - \zeta_k$ and the iteration counter k :

$$(32) \quad \begin{bmatrix} I - D_\xi \Phi_1 & D_\zeta \Phi_1 \\ D_\xi \Phi_2 & I - D_\zeta \Phi_2 \end{bmatrix} \begin{bmatrix} \Delta \xi_k \\ \Delta \zeta_k \end{bmatrix} = - \begin{bmatrix} \xi_k - \Phi_1(\xi_k, \zeta_k) \\ \zeta_k - \Phi_2(\xi_k, \zeta_k) \end{bmatrix}.$$

We only want to use the existing solvers, i.e. the iteration mappings Φ_1 and Φ_2 . In particular, we do not have direct access to the partial derivatives in Eq. (32). But if we solve the system Eq. (32) by an iterative method [36], all we need is a way to compute the product of the Jacobian matrix in Eq. (32) by an arbitrary vector.

We now want to consider a single iteration, and we will drop the iteration indices for ease of writing, and only look at computing the vector $[\Delta\xi, \Delta\zeta]^T$ from the right hand side. To start, we use — symbolically — block-Gauss elimination on the system Eq. (32):

$$(33) \quad \Delta\xi = -(I - D_\xi\Phi_1)^{-1}(\xi - \Phi_1(\xi, \zeta)) - C\Delta\zeta,$$

with

$$(34) \quad C = (I - D_\xi\Phi_1)^{-1}[D_\zeta\Phi_1].$$

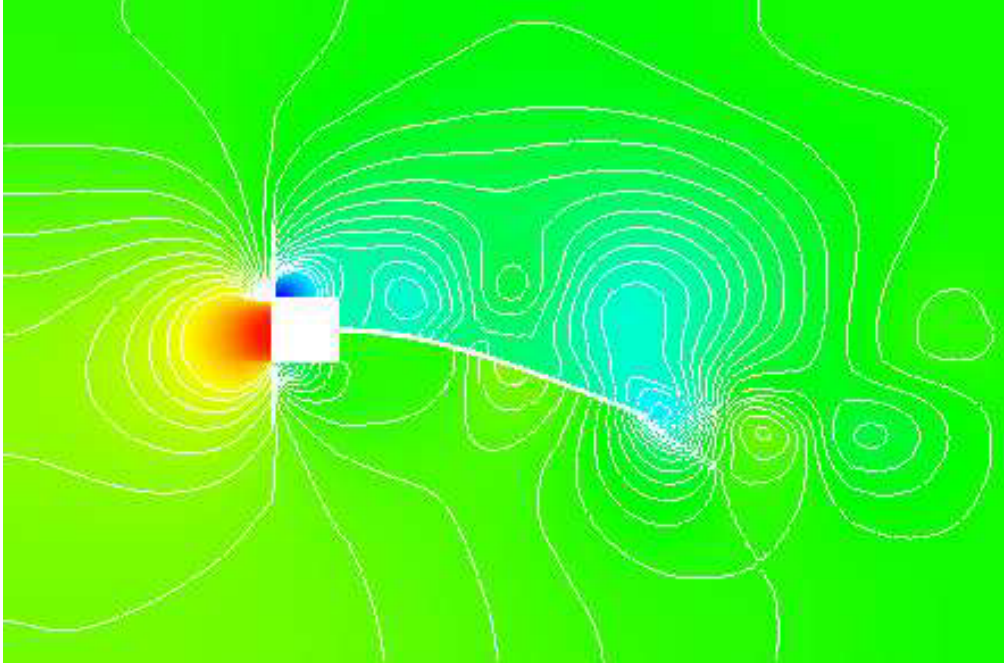


Figure 1: Vortex Shedding and Pressure Field

By inserting this into the second equation, we obtain

$$(35) \quad S\Delta\zeta = -r$$

with the Schur complement

$$(36) \quad S = I - [D_\zeta \Phi_2] - [D_\xi \Phi_2]C,$$

and

$$(37) \quad r = (\zeta - \Phi_2(\xi, \zeta)) + [D_\xi \Phi_2]q,$$

with

$$(38) \quad q = -(I - D_\xi \Phi_1)^{-1}(\xi - \Phi_1(\xi, \zeta)).$$

In this way we can solve the second equation for $\Delta\zeta$, and with this then solve

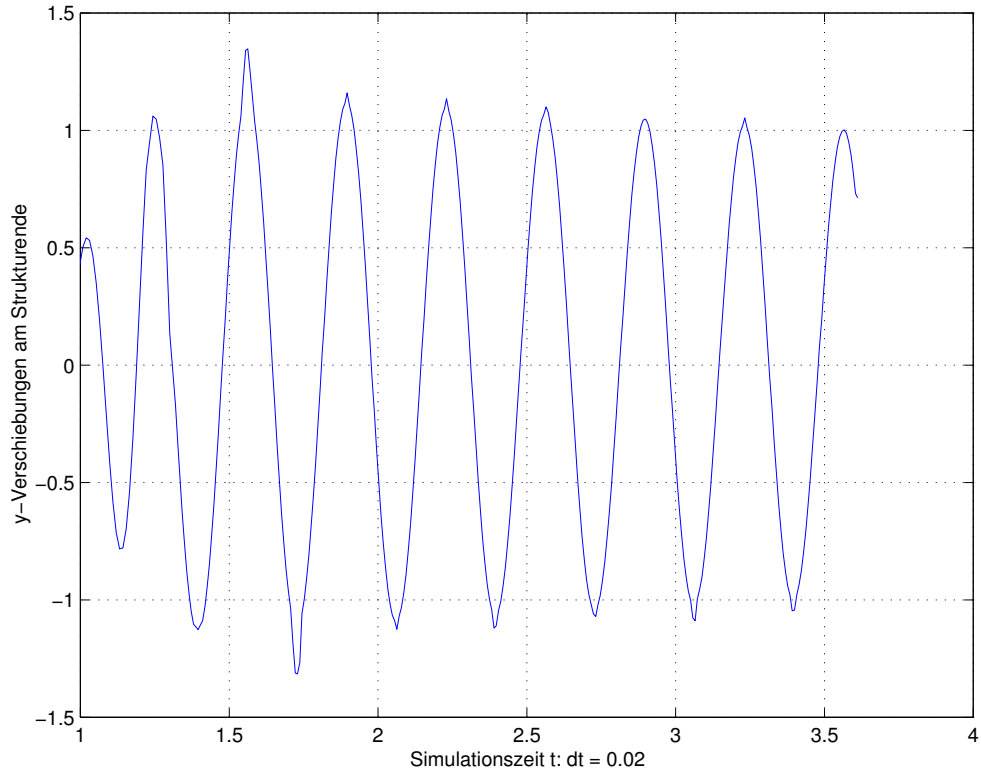


Figure 2: Displacement Response for Strong Coupling

the first equation for $\Delta\xi$. One step of this Newton-Raphson iteration may now be formulated as follows:

1. Solve $(I - D_\xi \Phi_1)q = \Phi_1(\xi, \zeta) - \xi$ for q .

2. With $r = \zeta - \Phi_2(\xi, \zeta) + [D_\xi \Phi_2]q$,
3. solve $S\Delta\zeta = -r$ for $\Delta\zeta$.
4. Compute $\Delta\xi = q - C\Delta\zeta$.

It remains to specify how the Jacobians are computed, and how the terms with matrices are handled. To solve the first equation, we use the iterative solver for the fluid, Φ_1 . One iteration there can also be seen as one Newton-Raphson step for the solution q of the equation $\xi - \Phi_1(\xi + q, \zeta) = 0$ when ξ and ζ are fixed. So with the iterative solver Φ_1 we obtain $q \approx z_m - \xi$, where

$$z_{j+1} = \Phi_1(z_j, \zeta), \quad j = 0, 1, \dots, m-1, \quad m > 1 \text{ with } z_0 = \xi.$$

To apply $[D_\xi \Phi_2]$ in the second step we use finite differences:

$$r = \zeta - \Phi_2(\xi, \zeta) + [D_\xi \Phi_2]q \approx \zeta - \Phi_2(\xi + q, \zeta).$$

In the third step an iterative Krylov method — here BiCGStab — is used to solve the system with the Schur complement matrix S ; and we only need the action of S on some other vector w , again approximated via finite differences with some (small) step-size h :

$$(39) \quad Sw = \frac{1}{h}S(hw) = \frac{1}{h}((I - D_\zeta \Phi_2)(hw) - [D_\xi \Phi_2]C(hw)) \\ \approx \frac{1}{h}(hw + \Phi_2(\xi - C(hw), \zeta + hw) - \Phi_2(\xi, \zeta))$$

Here we need $C(hw) =: v$, the solution of

$$(I - D_\xi \Phi_1)v = [D_\zeta \Phi_1](hw),$$

computed as in the first step, where additionally finite differencing is used for $[D_\zeta \Phi_1](hw)$:

$$[D_\zeta \Phi_1](hw) \approx \Phi_1(\xi, \zeta + hw) - \Phi_1(\xi, \zeta).$$

In the fourth step we know q , and again we need the action of C on some vector, this time $\Delta\zeta$, known from the third step. It is again done as in the third step for $C(hw)$.

Convergence of these approximative Newton methods has been investigated in [37, 35, 36, 34], and quadratic convergence has been shown under some assumptions on the subsystem solvers.

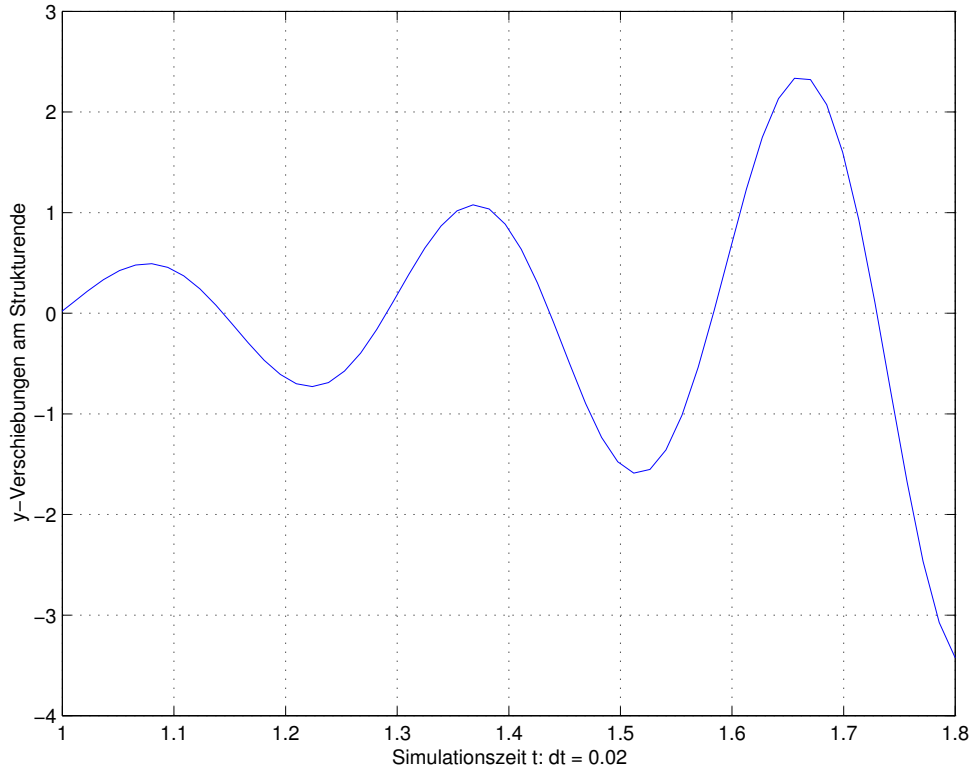


Figure 3: Displacement Response for Weak Coupling

7 Numerical Results

The numerical methods presented so far will be demonstrated [22] on a little two-dimensional example shown in Fig. 1, which is nevertheless quite challenging, and was introduced in [38]. It is a rigid block with a thin elastic appendage, in an incompressible but viscous flow from the left. The bluff shape of the square block induces vortices, which develop along the elastic appendage. Although the setup is symmetric, the vortices start to develop unsymmetrically, alternating from the top and from the bottom. As there are strong pressure variations in the vortices, this interacts with the elastic appendage, which in turn starts oscillating quite strongly. This is the situation shown in Fig. 1, together with the pressure field in the flow.

In Fig. 2 we show the response of the tip of the elastic appendage. This was computed with the strong coupling algorithm introduced in section 5 and section 6. If we solve this problem with the same time step but with an explicit staggering or so called weak coupling method, we observe the response in Fig. 3,

and one immediately recognises the instability of the numerical scheme.

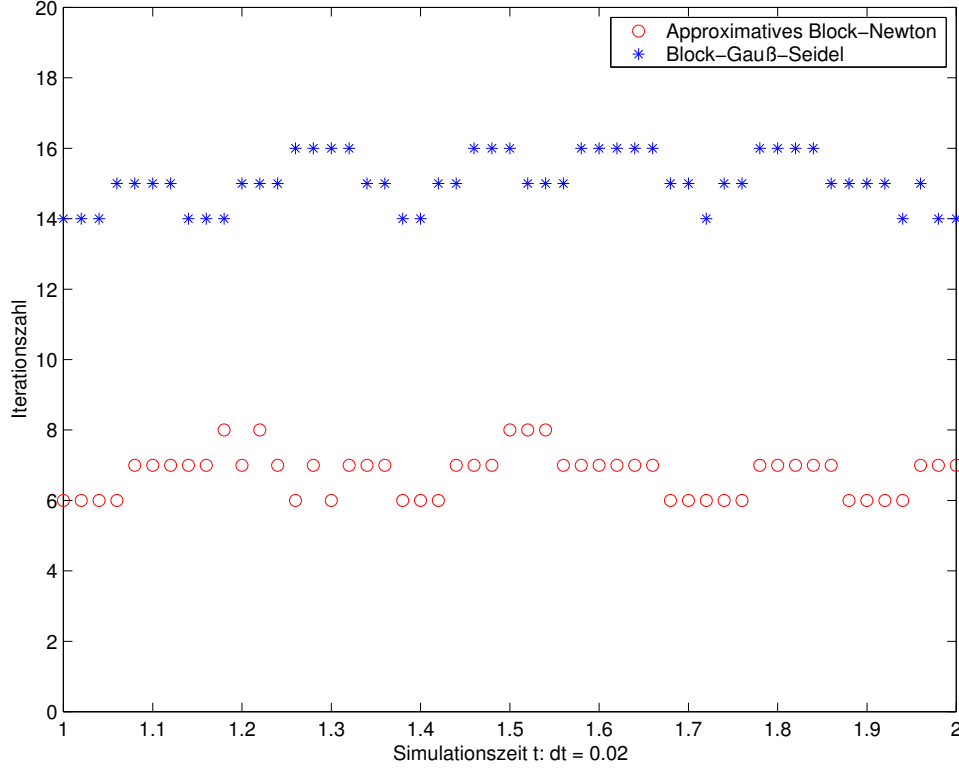


Figure 4: Iteration Count for block-Gauss-Seidel and block-Newton

In another test, we compare the block-Gauss-Seidel method as described in section 5 with the block-Newton method from section 6. For the same configuration as before, we compare the number of iterations of either method in each time step in Fig. 4. The superior convergence characteristic of the block-Newton method is obvious. But as the block-Newton method is more expensive to perform per iteration, this does not necessarily translate into an overall performance gain — of course apart from those cases where the block-Newton converges and the block-Gauss-Seidel method does not. Therefore, as a measure of numerical expenditure, we count the number of times the subsystem solvers have to be called per time step. This is shown in Fig. 5, and again the advantage is on the block-Newton side.

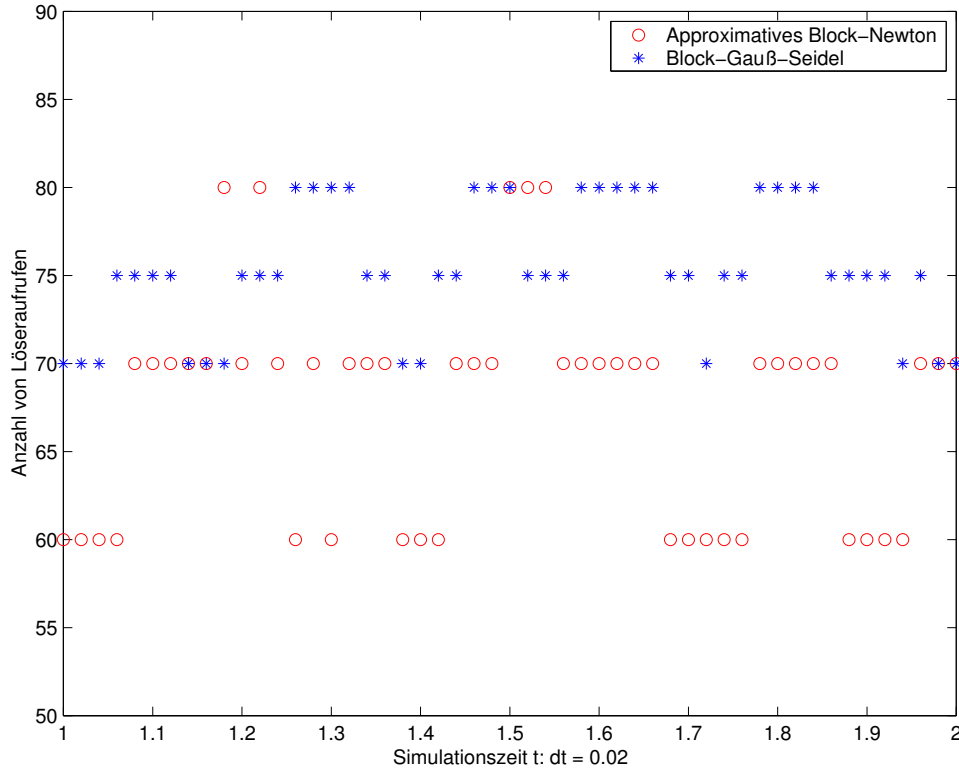


Figure 5: Count of Subsystem Solver Calls for block-Gauss-Seidel and block-Newton

8 Conclusion

For the problem of fluid-structure interaction, we have proposed a partitioned method to compute the transient response. We have proposed a general framework to formulate this coupling problem as a differential algebraic equation (DAE). For the sake of stability, implicit time stepping methods are considered.

The global system to be solved in each time step is approached with the solvers for the individual subsystems. The nonlinear block-Jacobi and nonlinear block-Gauss-Seidel methods come very naturally. We have discussed the problems which arise with this approach. We propose to solve the global implicit equations with an approximate block-Newton method. The linear system which arises in each iteration is solved via the subsystem solvers and a Krylov iterative method.

This iterative method only requires the possibility to use the subproblem solvers in an iteration-by-iteration fashion. The computation of the derivatives is approximated by the subproblem solvers and by finite differencing. These methods are faster both in number of iterations and in numerical effort; additionally

they have superior convergence characteristics. Needless to say that the general ideas about formulating and solving coupling problems are not specific to fluid-structure interaction and can be used in other circumstances as well.

References

- [1] Morand H, Ohayon R. Fluid-Structure Interaction. Chichester: John Wiley & Sons; 1995.
- [2] Farhat C. Parallel and distributed solution of coupled nonlinear dynamic aeroelastic response. In: Papadrakakis M, editor. Parallel Solution Methods in Computational Mechanics. Chichester: John Wiley & Sons; 1997.
- [3] Rifai S, Johan Z, Wang W-P, Grisval J-P, Hughes TJR, Ferencz M. Multi-physics simulation of flow induced vibrations and aeroelasticity on parallel computing platforms. *Comp Meth Appl Mech Eng* 1999; 174:393–417.
- [4] Pahlke K. Berechnung von Strömungsfeldern um Hubschrauberrotoren im Vorwärtsflug durch die Lösung der Euler-Gleichungen. PhD thesis 1998; Technische Universität Braunschweig.
- [5] Hierholz K-H. Ein numerisches Verfahren zur Simulation der Strömungs-Struktur-Interaktion am Hubschrauberrotor, PhD thesis 1999; Universität Stuttgart.
- [6] Mouro J, LeTallec P. Fluid structure interaction with large structural displacement. *Comp Meth Appl Mech Eng* 2001; 190:3039–3067.
- [7] Dubini G, Pietrabissi R, Montevocchi FM. Fluid-structure interaction in bio-fluid mechanics. *Med Eng Phys* 1995; 17(2):609–617.
- [8] Blom FJ. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Comp Meth Appl Mech Eng* 1998; 167:369–391.
- [9] Piperno S, Farhat C, Larroustou B. Partitioned procedures for the transient solution of coupled aeroelastic problems. *Comp Meth Appl Mech Eng* 1995; 124:79–112.
- [10] Piperno S. Explicit/implicit fluid-structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *Int J Num Meth Fluids* 1997; 25:1207–1226.

- [11] Mok DP, Wall WA. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In: Wall WA, Bletzinger K-U, Schweizerhof K, editors. Trends in Computational Structural Mechanics. Barcelona: CIMNE, 2001.
- [12] Bathe KJ, Zhang H, Wang MH. Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. Computers & Structures 1995; 56:193–213.
- [13] Bathe KJ. Finite Element Procedures. Englewood Cliffs, NJ: Prentice-Hall; 1996.
- [14] Bathe KJ, Zhang H, Ji SH. Finite element analysis of fluid flows fully coupled with structural interactions. Computers & Structures 1999; 72:1–16.
- [15] Rugonyi S, Bathe KJ. On the analysis of fully-coupled fluid flows with structural interactions-a Coupling and Condensation Procedure. Int J of Comp Civil and Struct Engineering 2000; 1:29–41.
- [16] Zhang H, Bathe KJ. Direct and iterative computing of fluid flows fully coupled with structures. In: KJ Bathe, editor, Computational Fluid and Solid Mechanics, Amsterdam: Elsevier, 2001.
- [17] Rugonyi S, Bathe KJ. On finite element analysis of fluid flows fully coupled with structural interactions. Computer Modeling and Simulation in Engineering (CMES) 2001; 2:195–212.
- [18] Steindorf J, Matthies HG. Efficient partitioned methods for the computation of fluid-structure interaction on parallel computers. In: Topping BHV, editor. Proceedings of the Third Euro-Conference on Parallel and Distributed Computing for Computational Mechanics. Edinburgh: Civil-Comp Press, 1999.
- [19] Codina R, Cervera M. Block-iterative algorithms for nonlinear coupled problems. In: Papadrakakis M, Buggeda G, editors, Advanced Computational Methods in Structural mechanics. Barcelona: CIMNE, 1996.
- [20] Park KC, Felippa CA. Partitioned analysis of coupled systems. In: Belytschko T, Hughes TJR, editors, Computational Methods in Transient Analysis. Amsterdam: North-Holland, 1983.
- [21] Arnold M, Günther M. Preconditioned dynamic iteration for coupled differential-algebraic systems. BIT Numerical Mathematics 2001; 41:1–25.
- [22] Steindorf J. Partitionierte Verfahren für Probleme der Fluid-Struktur Wechselwirkung, PhD thesis 2002; Technische Universität Braunschweig.

- [23] Matthies HG, Steindorf J. Efficient iteration schemes for non-linear fluid-structure interaction problems, In: BHV Topping, editor, Computational Mechanics: Techniques and Developments. Edinburgh: Civil-Comp Press, 2000.
- [24] Matthies HG, Steindorf J. How to make weak couplings strong, In: KJ Bathe, editor, Computational Fluid and Solid Mechanics, Amsterdam: Elsevier, 2001.
- [25] Matthies HG, Steindorf J. Fully coupled fluid-structure interaction using weak coupling. Proceedings in Applied Mathematics and Mechanics 2002; 1(1):37–38.
- [26] S Artlich, W Mackens. Newton-coupling of fixed point iterations. In: W Hackbusch, G Wittum, editors, Numerical Treatment of Coupled Systems. Braunschweig: Vieweg, 1995.
- [27] Donea J. Arbitrary Lagrangian-Eulerian finite element methods. In: Belytschko T, Hughes TJR, editors, Computational Methods in Transient Analysis. Amsterdam: North-Holland, 1983.
- [28] Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. AIAA Journal 1979; 17(10):1030–1037.
- [29] Grandmont C, Guimet V, Maday Y. Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid-structure interaction. In: Proceedings of the Second European Conference on Numerical Mathematics, Heidelberg: September 1997.
- [30] Batina JT. Unsteady Euler airfoil solutions using unstructured dynamic meshes. AIAA Journal 1990; 28(8):1381–1388
- [31] Farhat, C Lesoinne M. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional transient aeroelastic problems. Comp Meth Appl Mech Eng 2000; 182:499–515
- [32] LeTallec P, Mani S. Conservation laws for fluid-structure interactions. Technical Report CEREMADE 1999, Université de Paris Dauphine.
- [33] Arnold M. Constraint partitioning in dynamic iteration methods. submitted to Zeitschr für Angew Math und Mech 2000.
- [34] Mackens W, Menck J, Voss H. Numerical coupling of subsystems. Zeitschr für Angew Math und Mech 1999; 79(S3):S871–S872.

- [35] Miekkala U, Nevanlinna O. An approximate Newton method for coupled nonlinear systems. *SIAM J Sci Stat Comput* 1987; 8:459–482.
- [36] Mackens W, Menck J, Voss H. Numerical system synthesis: Concepts for coupling subsystem solvers. Technical Report 1998, Technische Universität Hamburg-Harburg.
- [37] Chan TF. An approximate Newton method for coupled nonlinear systems. *SIAM J Numer Anal* 1985; 22(5):904–913.
- [38] Wall WA, Ramm E. Fluid-structure interaction based upon a stabilized (ALE) finite element method. In: *Proceedings of the 4th World Congress on Computational Mechanics — New Trends and Applications*, Barcelona: CIMNE, 1998.

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 1998-02

1998-02	S. Petri, M. Bolz, H. Langendörfer	Transparent Migration and Rollback for Unmodified Applications in Workstation Clusters
1998-03	M. Cohrs, E. H. A. Gerbracht, W. Struckmann	DISKUS - Ein Programm zur symbolischen Diskussion reeller elementarer Funktionen
1998-04	C. Lindig	Analyse von Softwarevarianten
1998-05	Gregor Snelting, Frank Tip	Reengineering Class Hierarchies Using Concept Analysis
1998-06	Juliana Küster Filipe	On a Distributed Temporal Logic for Modular Object Systems
1998-07	J. Schönwälder, M. Bolz, S. Mertens, J. Quittek, A. Kind, J. Nicklisch	SMX - Script MIB Extensibility Protocol Version 1.0
1998-08	C. Heimann, S. Lauterbach, T. Förster	Entwurf und Implementierung eines verteilten Ansatzes zur Lösung langrechnender Optimierungsprobleme aus dem Bereich der Ingenieurwissenschaften
1999-01	A. Zeller	Yesterday, my program worked. Today, it does not. Why?
1999-02	P. Niebert	A Temporal Logic for the Specification and Verification of Distributed Behaviour
1999-03	S. Eckstein, K. Neumann	Konzeptioneller Entwurf mit der Unified Modeling Language
1999-04	T. Gehrke, A. Rensink	A Mobile Calculus with Data
2000-01	T. Kaiser, B. Fischer, W. Struckmann	The Modula-2 Proving System MOPS
2000-02	J. Saperia, J. Schönwälder	Policy-Based Enhancements to the SNMP Framework
2000-03	A. Casties	Finite-Element-Interpolation der rumlichen Dichten eines Vielteilchensystems auf ungeordneten Gittern
2000-04	J. Koslowski	A 2-dimensional view of the Chu-construction
2000-05	S. Eckstein, P. Ahlbrecht, K. Neumann	Von parametrisierten Spezifikationen zu generierten Informationssystemen: ein Anwendungsbeispiel
2000-06	F. Strauß, J. Schönwälder, M. Mertens	JAX - A Java AgentX Sub-Agent Toolkit
2000-07	F. Strauß	Advantages and Disadvantages of the Script MIB Infrastructure
2000-08	T. Gehrke, U. Goltz	High-Level Sequence Charts with Data Manipulation
2000-09	T. Firley	Regular languages as states for an abstract automaton
2001-01	K. Diethers	Tool-Based Analysis of Timed Sequence Diagrams
2002-01	R. van Glabbeek, U. Goltz	Well-behaved Flow Event Structures for Parallel Composition and Action Refinement
2002-02	J. Weimar	Translations of Cellular Automata for Efficient Simulation
2002-03	H. G. Matthies, M. Meyer	Nonlinear Galerkin Methods for the Model Reduction of Nonlinear Dynamical Systems
2002-04	H. G. Matthies, J. Steindorf	Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction
2002-05	H. G. Matthies, J. Steindorf	Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction
2002-06	H. G. Matthies, J. Steindorf	Strong Coupling Methods